L 77

中华人民共和国国家标准

GB/T 15532—1995

计 算 机 软 件 单 元 测 试

Computer software unit testing

1995-04-05发布 1995-12-01实施

国 家 技 术 监 督 局 发 布

中华人民共和国国家标准

GB/T 15532—1995

计算机软件单元测试

Computer software unit testing

1 主题内容与适用范围

1.1 主题内容

软件单元测试是一个过程。本标准为该过程规定了一个标准的方法，使之成为软件工程实践中的基础。该方法是一种综合的方法，目的是对软件单元进行系统化的测试，包括测试计划的执行、测试集的获取以及测试单元与其需求的对照衡量。对照衡量包括使用样本数据来执行被测单元，并将该单元的实际结果与单元的需求文件中指定的结果进行比较。

本标准描述了一个测试过程，它由一系列具有层次结构的阶段、活动及任务组成，且为每一活动定义了一个最小任务集。

1.2 适用范围

本标准可适用于任何计算机软件的单元测试(包括新开发的或修改过的软件单元)。本标准并不规定这些软件的类型，也不规定哪些软件必须进行单元测试。

本标准不涉及其他综合性的单元验证或确认过程，象评审(例如走查、审查)、静态分析(例如一致性核查、数据流分析)或形式化分析(例如正确性证明、符号执行)。

本标准不要求使用特定的测试机制或工具。本标准也不蕴含任何特定的方法学以进行文件控制、配置管理、质量保证、或测试步骤管理。同时也不规定软件排错的过程。

本标准的使用者可以是测试人员，也可是开发人员。

2 引用标准

GB 9386　计算机软件测试文件编制规范

GB/T 11457　软件工程术语

GB/T 12505　计算机软件配置管理计划规范

3 术语

下列术语定义适用于本标准，其他术语见 GB 9386 和 GB/T 11457。

3.1 特性　characteristic

见数据特性(3.2 条)或软件特性(3.5 条)。

3.2 数据特性　data characteristic

数据的一种固有的(也可能是非固有的)性质、质量或特征(例如数据使用率、格式、值范围或域值间关系)。

3.3 非过程性编程语言　monprocedure programming language

与过程性编程语言相对。是一种用于表达问题的参数，而不是表达解决问题的步骤的计算机编程语言(例如：报告生成器或分类的规范化语言)。

国家技术监督局1995-04-05批准　　1995-12-01实施

3.4 过程性编程语言 procedure programming language

与非过程性编程语言相对。是一种用于表达操作步骤，以供计算机执行的编程语言（例如COBOL）。

3.5 软件特性 software characteristic

软件的一种固有的（也可能是非固有的）性质、质量或特征（例如功能、性能、属性、设计约束、状态数目、分支的行数等）。

3.6 软件特征 software feature

由需求文件所规定或蕴含的软件特性（例如：功能、性能、属性或设计约束）。

3.7 软件测试事件 software test incident

在软件测试期间所发生的任何事件。

3.8 状态数据 state date

确定测试单元内部状态的数据，它用于建立状态或与现存状态比较。

3.9 测试对象 test objective

在指定条件下，通过对软件的实际状况与软件文件中所描述的状况进行比较来测量的软件特征集。

3.10 测试集结构 test set architecture

测试用例集（测试集）的嵌套关系，它能直接反映测试对象的层次分解情况。

3.11 测试单元 test unit

一个包括一个或多个计算机程序模块及相应控制数据（例如表格）、调用过程、操作过程的模块集合，且该集合成员满足下列条件：

a. 所有模块属于同一个计算机程序系统；

b. 集合中至少有一个模块（新的或改变过的模块）尚未完成单元测试；

c. 所有模块及相应数据和过程的集合是一个测试过程的唯一对象。

注：① 一个测试单元可能出现在从一个单独的模块到一个完整的程序这样一种设计层次的任何一个级别中。因此，一个测试单元可能是一个模块、一些模块，或一个具有相关数据和过程的完整的计算机程序。

② 一个测试单元可能包含一个或多个已进行过单元测试的模块。

3.12 单元 unit

见测试单元。

3.13 单元需求文件 unit requirement documentation

论述被测单元的功能需求、接口需求、性能需求及设计约束需求的文件。

4 单元测试活动

本章规定单元测试过程所涉及的活动，每个活动按输入、任务和输出这样的结构加以描述。所描述的阶段及活动如下：

a. 完善测试计划：

——制定方法、资源及进度的计划；

——确定需测试的与需求有关的特性；

——细化计划。

b. 获得测试集：

——设计测试集；

——执行计划及实现设计。

c. 评价测试单元：

——执行测试规程；

——核对终止情况；

——评价测试效果和测试单元。

所有活动的流程见图 1。

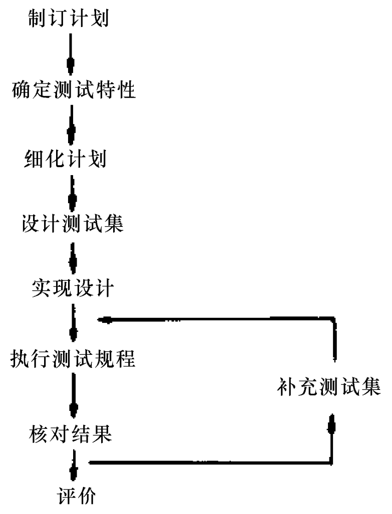

图 1　单元测试活动流程

当一个以上的单元需进行单元测试时(例如所有的这些单元均与一个软件项目有关),则计划活动须指出每个单元在整个测试单元集合中的位置,以免在每个测试单元中重复。

在一般情况下,除了图 1 中执行测试规程和核对结果这两个循环活动外,所有活动必须顺序进行。对于除制订计划阶段外的任何一个活动,若其前面的活动或某一外部事件(例如进度、需求、设计)有错,则有必要重新执行其前面的若干个活动,然后返回到当前活动。

各阶段的输入、输出数据流见图 2。

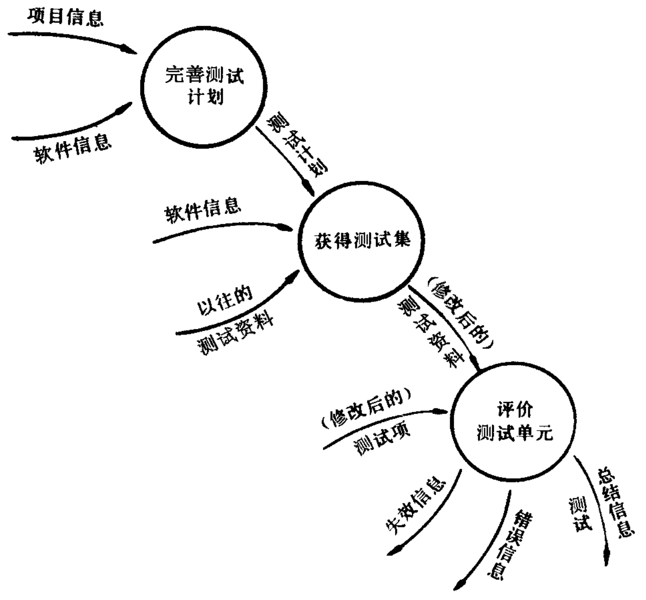

图 2　软件单元测试各阶段的主要数据流

在每个阶段,每个基本活动都连有其自身的输入集和输出集,其内容由一系列任务组成。本标准描

述了每个活动的输入、任务、输出。所有活动的输出集应当包含足够的信息来创建至少以下两个文件：一份测试设计说明及一份测试总结报告。所有文件必须符合 GB 9386 中的规定。所有的测试文件必须标明作者及日期。

测试设计说明将从确定测试特性、细化计划及设计测试集这几个活动中获得信息；测试总结报告将从所有的活动中获得信息。

4.1 制订方法、资源及进度的计划

总的单元测试计划应当在综合测试计划期间制订，且应在相应的计划文件中作出记录。

4.1.1 输入

a. 项目计划；

b. 软件需求文件。

4.1.2 任务

a. 指定单元测试的总方法

确定测试欲发现的风险区域。指定对确定特性(例如需测试的特性)、设计测试集或实现测试(例如必须使用的测试集)等这几个活动阶段的限制。

确定现有的输入、输出和数据资源(例如测试文件、制作文件、测试数据生成器)，确定数据确认的总技术。确定用于记录、收集、化简和确认输出数据的总技术。描述与被测试的单元有直接接口的应用软件的准备情况。

b. 指定完备的测试要求

确定单元测试集所覆盖的区域(例如软件特征、过程、状态、功能、数据特性、指令等)以及对每一区域所要求的覆盖程度。

在软件开发期间进行单元测试时，每一软件特征必须至少被一测试用例所覆盖，例外情况须得以批准。此原则也适用于软件维护时的单元测试。

当在软件开发期间测试一个用过程性语言(例如 COBOL)实现的单元时，对每一指令(能够到达及执行的)，除非该指令所在的模块已经独立地进行过单元测试，或者得到某种特许，它必须被某一测试用例所覆盖。此原则也适用于软件维护时用过程性语言实现的软件的单元测试。

c. 指定终止测试的要求

指定单元测试过程正常终止的需求。终止需求必须满足需求完备性。

确定会导致单元测试过程异常终止的任何情况(例如发现主要的设计缺陷，到达的最终期限)以及确定其相应的通告过程。

d. 决定资源的要求

估计进行测试集获取、初始启动及后续测试活动反复执行所需的资源。应考虑硬件情况、访问时间(例如所用的计算机时间)、通信或系统软件、测试工具、测试文件等。

确定需要准备的以及各部门响应所需的资源，包括那些对于其交付时间有严格要求的资源(例如定制的测试工具)，并安排这些资源。

确定对单元测试及单元排错负责的部门、人员技能、数量及可参加时间的要求。

e. 指定总的进度安排

指定由资源和测试单元所决定的单元测试活动的进度。

4.1.3 输出

a. 单元测试计划(从 4.1.2 条的 a～e 得到)；

b. 单元测试的总体资源请求(若能从 4.1.2 条的 d 得到)。

4.2 确定需测试的与需求有关的特性

4.2.1 输入

a. 单元需求文件；

b. 软件结构设计的文件(若需要)。

4.2.2 任务

a. 研究功能需求

研究单元需求文件中描述的每一功能。保证每一功能有唯一的标识符。若需要的话,应对需求进行分类。

b. 确定附加需求及相应规程

对于那些没有被需求指定,却在单元测试一级有效测试的软件特性(例如软件性能、属性或设计约束),确定与之相关的需求语句,使之成为附加需求。确定那些仅与待测试单元有关的使用或操作规程。确保每一附加需求及规程有唯一的标识符。若需要的话,应对需求进行分类。

c. 确定单元状态

若单元需求文件指定或蕴含了多种状态(例如不活动、等待接收、处理)软件,则确定每一状态及每一有效状态转换。保证每一状态及状态转换有唯一标识符,若需要的话,应对需求进行分类。

d. 确定输入及输出数据特性

确定待测试单元的输入及输出数据结构。对每一结构,确定其特性,诸如使用率、格式、值范围和域值之间的关系。对每个特性,指定其有效范围。保证每一特性有唯一标识符。若需要的话,应对需求进行分类。

e. 选择包含于测试中的各要素

选择待测试的软件特征。选择其相应规程、状态及状态转换,以及测试时的有关数据特性。无效及有效数据都应选择。当无法进行这种完整的测试时,则应该利用如何使用该单元的信息决定选择的内容。对于不能选择的要素,确定由此可能带来的风险问题。

将所选的特性、规程、状态、状态转换及数据特性等数据记录在单元测试设计说明中的"被测试的特性"一章中(见 GB 9386)。

4.2.3 输出

a. 测试过程中包含的各要素的列表(从 4.2.2 条的 e 得到);

b. 单元需求分类的信息(若能从 4.2.2 条的 a～d 得到)。

4.3 细化计划

4.3.1 输入

a. 测试过程中包含的各要素的列表(从 4.2.2 条的 e 得到);

b. 单元测试计划(从 4.1.2 条的 a～e 得到)。

4.3.2 任务

a. 方法

确定可以考虑利用的现有的测试用例及测试规程。确定用于数据确认的任何特定技术。确定用于输出记录、输出收集、输出化简及输出确认所用的技术。将细化的方法记录于单元的测试设计说明文件中的"方法详述"一章中(见 GB 9386)。

b. 详述指定的资源需求

确定所指定的测试单元所需的资源(例如与该单元直接接口的软件)。并为已确定的资源作准备。将指定资源的需求记录在单元测试设计说明的"方法详述"一章中。

c. 指定详细进度

根据支撑软件、指定资源、所使用单元的可获得性及组装进度,为单元测试规定相应进度。将该进度记录于单元的测试设计说明的"方法详述"一章中。

4.3.3 输出

a. 详细的单元测试计划(从 4.3.2 条的 a～c 得到);

b. 单元测试的指定资源要求(若能从 4.3.2 条的 b 得到)。

4.4 设计测试集

4.4.1 输入

a. 单元需求文件；

b. 测试过程中所包含的各要素的列表(从 4.2.2 条的 e 得到)；

c. 单元测试计划(从 4.1.2 条的 a 和 b 及 4.3.2 条的 a 得到)；

d. 单元设计文件；

e. 来自以前测试的测试规格说明(若可获得的话)。

4.4.2 任务

a. 设计测试集的层次结构

根据待测试的软件特征和由所选的有关要素(例如规程、状态转换、数据特性)所指定或蕴含的情况,设计一个按层次分解好的测试对象集,使得最低层的每一对象能直接用一些测试用例进行测试。选择合适的现有的测试用例,将测试用例标识符组与最低层的相应的对象相关联。将对象层次和相应的测试用例标识符记录于单元的测试设计说明中的"测试用例名称"一章中(见 GB 9386)。

b. 按需求获得清晰的测试规程

单元需求文件、单元测试计划及测试用例说明的组合可能会隐含地指定出单元测试规程,从而不需要更细致的"测试规程说明"。选择现存的测试规程,稍作修改或不加修改地使用。

若单元测试设计说明的补充章条有要求,或另外的规程说明文件有要求,应指定相应的附加的规程。每一种选择都应与 GB 9386 相吻合。当测试用例和测试规程的对应关系不是很明显时,用表格连接它们,并将其放于单元测试设计说明中。

c. 获得测试用例说明

指定新的测试用例,可参考现存的测试用例说明。

将该测试用例直接记录于或通过引用的方式记录于单元的测试设计说明的补充章条中或另外的文件中。记录的文件必须符合 GB 9386 的要求,并放于单元的测试设计说明中。

d. 根据设计信息,按需要扩大测试用例集的说明

根据单元设计的信息,按需要更新测试集层次结构,注意应与 4.4.1 条的 a 保持一致,并考虑所选算法及内部数据结构等软件特征。

如果要确定控制流程及确定必须记录的内部数据的变化情况,则应考虑到可能产生的特殊记录的困难。例如,跟踪复杂算法中的控制流或跟踪内部数据结构(如栈或树)的变化时存在的记录困难。若需求的话,应增强单元设计(例如格式化数据结构、转储功能)以增强单元的可测试性。

根据单元设计中的信息,描述那些新增加的测试用例,并完成各部分的测试用例说明,同时应与 4.4.2 条的 c 保持一致。

e. 完成测试设计说明

完成被测单元的测试设计说明,并与 GB 9386 相一致。

4.4.3 输出

a. 单元测试设计说明(从 4.4.2 条的 e 得到)；

b. 附加的测试规程说明(若能从 4.4.2 条的 b 得到)；

c. 附加的测试用例说明(若能从 4.4.2 条的 c～d 得到)；

d. 单元设计的增强需求(若能从 4.4.2 条的 d 得到)。

4.5 执行计划及实现设计

4.5.1 输入

a. 单元测试计划(从 4.1.2 条的 a、c、e 及 4.3.2 条的 a～c 得到)；

b. 在单元测试设计说明或附加文件中的测试用例说明(从 4.4.2 条的 c、d 得到)；

c. 软件数据结构描述；

d. 测试支持资源；

e. 测试项；

f. 来自以前测试活动的测试数据(若存在)；

g. 来自以前测试活动的测试工具(若存在)。

4.5.2 任务

a. 获得并验证测试数据

对于能稍作修改或不作修改便可使用的测试数据，获得它们的一份备份，按需求产生新的数据。为保证数据的一致性和完整性，还应包含附加数据。按照软件数据结构规格说明验证所有数据。当测试用例和数据集的关系不明显时，用表格来记录此种关系，并放于单元测试设计说明中。

b. 获得指定资源

获得 4.3.2 条的 b 中指定的测试支持资源。

c. 获得测试项

收集包含已有的手册、操作系统规程、控制数据(如表格)和计算机程序在内的所有测试项，获得在测试设计期间确定的与测试单元有直接接口的软件。

当测试一个用过程性语言实现的单元时，要保证执行轨迹信息足以能够满足基于代码的程序的完备性要求。

将每一项的标识符记录于单元测试总结报告的“简述”一章中(见 GB 9386)。

4.5.3 输出

a. 验证过的测试数据(从 4.5.2 条的 a 得到)；

b. 测试支持资源(从 4.5.2 条的 b 得到)；

c. 测试项的配置(从 4.5.2 条的 c 得到)；

d. 初步总结(从 4.5.2 条的 c 得到)。

4.6 执行测试规程

4.6.1 输入

a. 验证过的测试数据(从 4.5.2 条的 a 得到)；

b. 测试支持资源(从 4.5.2 条的 b 得到)；

c. 测试项的配置(从 4.5.2 条的 c 得到)；

d. 测试用例说明(从 4.4.2 条的 c、d 得到)；

e. 测试规程说明(若 4.4.2 条的 b 能够产生)；

f. 故障分析结果(从排错过程得到)。

4.6.2 任务

图 3 为执行测试规程活动内的控制流程图。

a. 运行测试

建立测试环境，运行测试集。在单元测试总结报告的“结果概述”一章中记录所有的软件测试事件。

b. 判定结果

对每一个测试用例，利用测试用例描述文件中有关的所需结果的规格说明，来判定单元测试活动是通过还是失效。将通过或失效结果记录于单元测试总结报告的“结果概述”一章中。将资源消耗数据记录于报告的“活动总结”一章中(见 GB 9386)。当测试一个用过程性语言实现的单元时，收集执行轨迹的总结信息，且将其添入总结报告中。

对每一次失效，应加以分析并将出错信息记录在测试总结报告的“结果概述”一章中，然后选择以下适用情况执行相应措施。

情况 1：测试规格说明或测试数据的故障

改正错误，将改正错误信息记录在测试总结报告的“活动总结”一章中，然后重新运行该测试。

情况 2:执行测试规程时的故障

重新运行未正确执行的规程。

情况 3:测试环境(例如系统软件)中的故障

将环境修正,将环境修正情况记录在测试总结报告的"活动总结"一章中,然后重新运行该测试;或者预先设置异常终止情况,将不能修正环境的理由记录于测试总结报告的"活动总结"一章中,然后开始核对终止情况(即开始执行 4.7 条的活动)。

情况 4:单元实现故障

修正错误,并将修正错误情况记录在测试总结报告的"活动总结"一章中,然后重新运行所有的测试;或者预先准备异常终止情况,将不能进行单元修正的理由记录于测试总结报告的"活动总结"一章中,然后开始核对终止情况(即开始执行 4.7 条的活动)。

情况 5:单元设计故障

修正单元设计并实现,在适当的时候修改测试规格说明及数据,将错误修正情况记录于测试总结报告的"活动总结"一章中,然后重新运行所有的测试。或者,预先设置异常终止情况,将不能进行设计修正的理由记录于测试总结报告的"活动总结"一章中,然后开始核对终止情况(即开始执行 4.7 条的活动)。

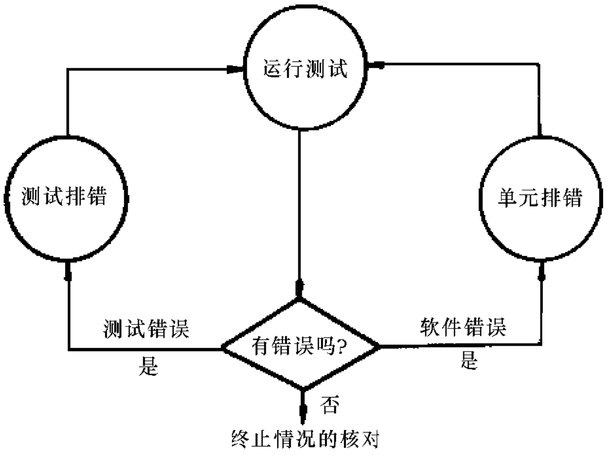

图 3　执行测试规程活动内的控制流程

4.6.3　输出

a.　包含在测试总结报告中的执行信息,其中包括测试输出、软件测试事件描述、故障分析结果、错误修正活动、不能修改错误的理由、资源消耗数据;对过程性语言实现的程序而言,还应包括执行轨迹总结信息(从 4.6.2 条的 a、b 产生);

b.　修订后的测试规格说明(若能从 4.6.2 条的 b 得到);

c.　修订后的测试数据(若能从 4.6.2 条的 b 得到)。

4.7　核对终止情况

4.7.1　输入

a.　完备性和终止情况的需求说明(从 4.1.2 条的 b、c 得到);

b.　执行信息(从 4.6.2 条的 a、b 得到);

c.　测试规格说明(从 4.4.2 条的 a～c 得到)(若有需要);

d.　软件数据结构描述(若有需要)。

4.7.2　任务

图 4 为结果核对活动内的控制流程图。

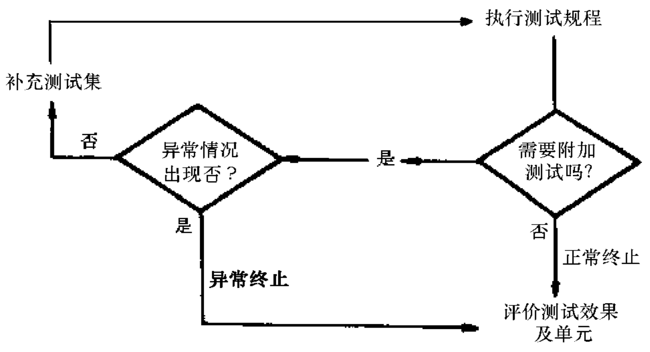

图 4 结果核对活动内的控制流程

a. 对测试过程的正常终止情况进行核对

根据完备性要求或失效记录，决定是否要增加新的测试。对于用过程性语言实现的程序，要分析执行轨迹总结信息(例如变量、数据流)。

若不需附加测试，则将正常终止情况记录于测试总结报告的“活动总结”一章中，然后开始评价测试效果及被测单元(即开始执行 4.8 条的活动)。

b. 对测试过程的异常终止情况进行核对

若满足异常终止条件(例如重要错误不能修正、超时)，则应将导致终止的特殊条件记录于测试总结报告的“活动总结”一章中，同时也应记录未完成的测试及未被修正的错误，然后开始评价测试效果及被测单元(即开始执行 4.8 条的活动)。

c. 补充测试集

当需要附加的测试且异常终止情况不满足时，通过下列步骤来补充测试集。

——更新测试集的结构，并与 4.4.2 条的 a 一致，且根据 4.4.2 条的 c 获得相应的测试用例说明；

——按需要根据 4.4.2 条的 b，修改测试规程说明；

——根据 4.5.2 条的 a，获得附加测试数据；

——将附加内容记录于测试总结报告的“活动总结”一章中；

——执行附加的测试(即返回 4.6 条的活动)。

4.7.3 输出

a. 记录于测试总结报告内的核对信息，包括终止条件及任何测试用例的附加情况(从 4.7.2 条的 a～c 得到)；

b. 附加的或修订后的测试规格说明(若能从 4.7.2 条的 c 得到)；

c. 附加的测试数据(若能从 4.7.2 条的 c 得到)。

4.8 评价测试效果和被测单元

4.8.1 输入

a. 单元的测试设计说明(从 4.4.2 条的 e 得到)；

b. 执行信息(从 4.6.2 条的 a、b 得到)；

c. 核对信息(从 4.7.2 条的 a～c 得到)；

d. 附加的测试用例说明(若能从 4.4.2 条的 c、d 得到)。

4.8.2 任务

a. 描述测试状态

将测试计划和测试规格说明的变化情况记录于测试总结报告的“差异”一章中(见 GB 9386)。要说明每次变化的原因。

对异常终止情况，要确定未能被测试活动充分地覆盖的区域，且将理由记录于测试总结报告的“测

试充分性评价”一章内(见 GB 9386)。

确定未能解决的软件测试事件以及不能解决的理由,并记录于测试总结报告的“结果概述”一章中。

b. 描述单元状态

将通过测试所反映的单元与其需求文件之间的差异记录于测试总结报告的“差异”一章中。

将测试结果及所发现的错误情况同需求对照,评价单元的设计与实现,将评价信息记录于测试总结报告的“测试充分性评价”一章中。

c. 完成测试总结报告

根据 GB 9386,完成测试总结报告。

d. 保存测试文件

确保测试得到的成果的收集、组织和存储,以备调用及重用。这些成果包括测试设计说明、附加的测试用例说明、附加的测试规程说明、测试数据、测试用例的产生规程、测试驱动程序和桩模块以及测试总结报告。

4.8.3 输出

a. 测试总结报告(从 4.8.2 条的 c 得到);

b. 测试成果的收集存储(从 4.8.2 条的 d 得到)。

附　录　A
实现及使用指南
（参考件）

本附录包含使用本标准时有益的信息。因此建议在作出更为详细的计划之前先阅读本附录。

A1　标准的使用

本标准有以下作用：

a.　作为为确定当前的实践活动而比较的基础；

b.　作为修改当前实践活动的思路之源；

c.　作为当前实践活动的替代物。

A2　附加的测试需求

对每个项目，象附加测试文件（如测试日志）的数量，应当描述的细致程度及批准、复查的数量和类型等等这样的需求都应详细说明。有些因素，如单元的批评意见、读者需求或合同说明会经常影响这些需求，本标准将这种需求留给用户自己去描述，该描述可作为单独项目的需求，亦可作为某个组织的标准。若这些需求是某个项目特指的，则应在项目计划、质量保证计划、证明及验证计划或全局的测试计划中进行描述。

A3　附加的测试文件

一般认为，测试设计说明和测试总结报告中包含的信息是完成测试过程后得到的最小文件集合。另外，通过在这些文件中增加附加内容或增加额外的文件，GB 9386 中描述的测试文件集可以满足所要求的任何测试信息。

A4　认可及评审

如要求更多的控制，应考虑以下附加任务：

a.　在计划阶段的末尾认可总的方法；

b.　在确定测试特性阶段的末尾认可所确定的需求；

c.　在细化计划阶段的末尾认可所描述的计划；

d.　在设计测试集的末尾认可测试说明；

e.　在实现测试阶段的末尾认可测试准备情况；

f.　在评估阶段认可测试总结报告。

A5　审计

当描述控制需求时有必要考虑审计问题。因此，应当从测试评审中产生足够的测试文件及报告，以满足所要求的所有审计信息。

A6　配置管理

配置管理以软件需求、软件结构设计、软件数据结构及单元需求文件作为输入源。必须合理地管理这些输入文件，以便确保我们持有的是现行的信息，并且任何变动都能得到通知。

单元测试的最终文件也应进行配置管理。必须管理好这些输出文件以便进行全面而经济的测试。详见 GB/T 12505。

A7 确定基于需求的特性

对单元开发人员而言,当在测试中确定基于需求的元素(如特性、规程、状态转换、数据特征)的有效集合时,心理因素(如自信心、关于单元设计的详细知识)起了阻碍作用。通常,这样的"确定特性"过程应当由其他人来完成。

有以下几种方式完成这种活动:

a. 开发人员之间相互确定这些特性;

b. 开发人员之间完整地测试他人的代码,这带来的优点是至少两名开发人员将会熟悉每个单元的详细知识;

c. 组织一个独立的测试小组。项目的大小或软件的重要性可以决定是否适合组织这样的独立的测试小组。

若开发人员为自己的软件确定基于需求的元素,他们应当在软件设计开发之前完成这种确定工作。

A8 用户的参与

若在测试某一单元时需与用户交互进行(如菜单显示),则应请用户参与确定基于需求的元素的工作,这样效果会更好。在作测试计划时向用户询问其使用情况会发现非常有价值的信息。例如,通过询问可能明确相对重要的单元功能,从而确定出测试的重点。

A9 更强的代码覆盖需求

针对单元的重要性或单元需求说明与设计信息的不足(如在维护旧的软件时发生的情况),可以加强 4.1.2 条的 b 中描述的基于代码的覆盖需求。其中一种方式便是将指令 覆盖的要求增强到分支覆盖的要求(即要求遍历单元 中每一个分支)。

A10 代码覆盖工具

这里大力推荐一种在测试单元执行时记录源代码覆盖量的自动化方式。之所以使用自动化方式就是因为手工覆盖分析不可靠且不经济。使用代码探测及报告工具就是一种自动化方式的工具,该工具将软件探测器放置于源代码中,以后在运行测试用例时便可提供一份总结了数据及控制流信息的报告。该报告会指出未运行过的指令。有些工具还能指出未运行的分支。有些编译器也具备这种特点。

A11 测试过程的补充

为了评价并增强单元测试的有效性,建议在单元测试之后的步骤如组装测试、系统测试、产品 使用等活动中收集失效数据。然后分析这些数据以便确定那些本应被单元测试检查出却未被查出的错误。

A12 本标准的使用

实现一个新技术的过程,其本身就是一个需要计划、实现及评价效果的过程。为了成功地实现基于本标准的测试,测试人员必须开发一份实现策略并将本标准作适当裁剪。这两个活动必须反映出组织内部的文化背景及权威性。若要成功地完成一个长期项目,还需要专门的管理以及支持的政策、工具、培训和起动协议。

A13 标准的可实施性

本标准同许多好的软件工程实践有一致的定义。某些组织采用与此相似的实践活动而另外一些却完全不同。无论如何,对许多决定选择本 标准并适应本标准的组织而言,它都会有某种变化。这种变化涉及到新的政策及规程、新的工具以及新的培训程序。若标准与实际相差太大,则有必要对标准进行某

种改变。解决实用性问题的答案从根本上讲是满足需要的问题。

附 录 B
概 念 及 假 定
（参考件）

B1 软件工程概念

本标准中描述的标准化单元测试过程是建立在软件工程一些基本概念之上的，B1.1～B1.8 条描述了这些概念。

B1.1 测试与验证、确认的关系

测试只是一系列包含验证、确认等活动中的一个。其他活动有技术评审（如代码审查），静态分析及正确性证明。综合性验证及确认过程的规范不属于本标准的范围。

B1.2 像产品开发般的测试

测试是一个开发产品的过程。其结果是产生一个测试集，包括使用的数据、测试支持软件及规程。该产品以文件形式记录于测试规范说明及报告中。同其他产品开发过程一样，开发测试集要求有计划、需求（测试目标）、设计、实现以及评价等阶段。

B1.3 排错过程的组成

排错过程由两个主要的活动组成。第一步活动即失效分析的目标是确定导致一次失效的所有错误的地点。第二步活动即改正错误的目标是去除所有查出的错误并避免产生新的错误。关于失效分析及错误改正的过程的规范说明不属于本标准的范围。

B1.4 测试与排错的关系

测试是为了检查错误而力图引出失效，而排错既要进行失效分析并判定有关错误的地址，又要改正错误。测试可能需要来自排错过程的失效分析的结果来决定终止测试、请求改变需求或进行错误改正等这样的活动。

B1.5 单元类型之间的关系

没有必要在设计单元、实现单元与测试单元之间建立一一对应的关系。几个设计单元可能组成一个实现单元（如一段程序），而几个实现单元可能组成一个测试单元。

B1.6 对设计与实践信息的需求

一般来讲，尽管测试的本质是将实际执行情况与所需求情况进行对照衡量，但并不能认为需求信息已足以帮助进行有效的测试。这是由于通常不可能测试所有可能的情况，而需求说明并不对失效率高的情况提供足够的指南。由于这些失效率高的情况是设计和实现选择时造成的结果，所以测试时通常需要设计和实现信息。

B1.7 测试中所考虑的元素说明的补充

在编制单元需求文件、单元设计文件以及在最后单元实现过程中，会渐渐发现更详细的关于测试单元的信息。其结果是，测试所考虑的元素可能会在不同的测试活动期间得到补充。

对过程性编程语言的实现而言（如 COBOL），元素说明出现三次补充。第一组是在确定特性这一活动阶段确定的，它基于单元需求文件；第二组是在设计测试集这一活动阶段确定的，它基于软件设计描述阶段所描述的单元设计信息（即算法与数据结构）；第三组是在核对结果这一阶段确定的，它基于单元代码。

对非过程性编程语言的实现而言（如报告生成器或分类的规格化语言），元素说明出现两次补充。第一次是在确定特性活动阶段，基于需求说明；第二次是在设计测试集活动阶段，基于非过程的规格说明书。

有一种补充方法,它允许在获得需求文件之时便开始单元测试,并最大限度地减少单元设计与单元代码的详细知识之间的差别。

B1.8 对创建测试设计说明的补充

测试设计说明中记录的信息是在确定特性、细化计划和设计测试集阶段收集的。随着每个测试活动阶段的进行,有关说明的相应章条记录了信息。所有的文件必须在设计测试集活动的最后重复的末尾完成。

B1.9 对创建测试总结报告的补充

测试总结报告中记录的信息是在测试计划阶段之外所有测试活动阶段得到的,在实现测试时初建,在执行及核对阶段修改,在评价阶段完成。

B2 测试假设

本标准中描述的单元测试的方法是建立在有关经济、心理、技术等几种假设之上的。B2.1～B2.7 条给出这种假设的重要性。

B2.1 单元测试的目标之一就是判断根据单元需求及设计文件而实现的单元的正确性与完备性。它试图在以下几项中发现错误:

- a. 单元需求特性,并结合其相应的描述(如不活动、活动等待一个信号、活动处理一个信号);
- b. 关于无效输入的单元手册;
- c. 仅与单元相关的任何使用及操作规程;
- d. 单元的算法或内部数据结构,或两者都有;
- e. 单元控制逻辑的判定边界。

B2.2 测试的内容之一是根据需求核对实际情况。人们总是非正式地说起接口测试、语句测试或需求说明测试,其意义就是对照相应的接口、语句、需求说明的描述来核对它们的实际情况。任何可核对的单元测试过程都必须有该单元的需求说明文件。本标准假设在测试开始之前该单元测试需求文件已经存在。

B2.3 单元需求说明文件必须经过完备性、可测试性及可跟踪性的全面评审。本标准假设需求说明已评审过,不论是作为评审过程的一个部分,还是以特殊的单元需求说明进行的评审。

B2.4 在查错的早期,往往有重要的经济效益。这意味着应当在一获得单元需求说明文件并进行测试之时便开始开发测试集,这是由于它会影响需求的验证及确认。这也意味着在单元一级应进行尽可能多的测试。

B2.5 项目测试的级别(如验收、系统、组装、单元)是在项目计划、验收及确认计划或全局测试计划中描述的。同时这些计划中也要描述适用于所有被测试单元的单元测试计划信息(如完备性需求、终止需求、资源总需求)。继而,根据对软件设计的分析,标识出测试单元,选择组装序列。

B2.6 完成一个任务所需要的输入条件和资源的可获得性是决定活动顺序和活动内的任务顺序的主要约束。若能获得必要的资源,某些活动及一个活动内的某些任务便可能得以并行执行。

B2.7 本标准认为拖延测试用例的设计是影响开销的最大因素,此处测试集是基于源代码特点和需求及设计的特点,这种影响直到测试集执行之后才结束。这样的方法使基于代码的设计任务最小化。若基于代码的设计在获得测试执行数据之前便已开始,则它应在那些基于需求及设计特点的测试用例已被说明之后再开始。

附加说明：

本标准由中华人民共和国电子工业部提出。
本标准由电子工业部标准化研究所归口。
本标准由上海计算机软件技术开发中心负责起草。
本标准主要起草人朱三元、刘光龙、冯惠、周庆隆、宿为民、陈淼芬。